Image Cover Sheet

National Defence Défense nationale

Canada

Defence and Civil
INSTITUTE OF ENVIRONMENTAL MEDICINE

INSTITUT DE MEDECINE ENVIRONNEMENTALE
pour la défense

1133 Sheppard Avenue West, PO Box 2000, North York, Ontario, Canada M3M 3B9
Tel. (416) 635-2000
Fax. (416) 635-2104

MAY 1994

DCIEM No 94-40

IMPLEMENTATION OF A HUMAN INFORMATION PROCESSING MODEL FOR TASK NETWORK SIMULATION

Keith C. Hendy

Defence and Civil Institute of Environmental Medicine
1133 Sheppard Ave. W., P.O. Box 2000
North York, Ontario
Canada M3M 3B9

TABLE OF CONTENTS

EXECUTIVE SUMMARY

IMPLEMENTATION OF A HUMAN INFORMATION PROCESSING MODEL FOR TASK NETWORK SIMULATION

DCIEM No. 94-40

Keith C. Hendy

Task network simulation is an analytical technique that is widely used to predict operator performance and/or workload during the early stages of systems design. Task network simulation is based on traditional time-line analysis methods, but allows the possibility of non-deterministic task characteristics such as completion times, sequences, outcomes etc. Many simulation environments allow task parameters to vary with various network states, which supports complex logical relationships, and time varying network behaviours.

The raw output from task network simulation represents a simulated timeline of the activity modelled. In general this is neither a measure of system performance nor operator workload. To provide this measure, a time-based metric of system performance or some model of the human information processor is required from which load, and eventually performance, can be inferred. While many approaches have been used to predict operator load and performance from simulated task timeline data few, if any, can claim a strong theoretical basis.

This report outlines the implementation of a theoretical framework for a new model of the human information processor for use in task network simulation. The development and validation of the Information Processing (IP) Model is described in detail elsewhere. This report deals only with those aspects that are necessary to take the ideas of the IP Model and adapt them for direct application to task network simulation. The material contained in this report provides the bridge between the conceptual descriptions of the IP Model, and the software requirements necessary to put that concept into practice. As part of this process, many parameters are defined and assigned tentative values so that the model can be run within the task network simulation environment.

In devising this implementation, many assumptions were made that are beyond the scope of earlier validation studies. Hence, further validation will be required to verify the model within the context of this implementation. As the specific purpose of the report is to describe the algorithmic and data base requirements for a specific software environment, some of the material is specific to that system. However, much of the material is of a more general nature and could be adapted to other software environments.

INTRODUCTION

Task network simulation is an analytical technique that is widely used (Hendy, 1994b) to predict operator performance and/or workload during the early stages of systems design. Task network simulation is based on traditional time-line analysis methods (Meister, 1985), but allows the possibility of non-deterministic task characteristics such as completion times, sequences, outcomes etc. Many simulation environments allow task parameters to vary with various network states, which supports complex logical relationships, and time varying network behaviours.

The raw output from task network simulation represents a simulated timeline of the activity modelled. In general this is neither a measure of system performance nor operator workload. To provide measure, a time-based metric of system performance or some model of the human information processor is required (Hendy, 1994a) from which load, and eventually performance, can be inferred. While many approaches have been used to predict operator load and performance from simulated task timeline data few, if any, can claim a strong theoretical basis.

Task network analysis is supported by a number of software simulation environments, for example: GASP and SLAM (Doering and Berheide, 1981); SAINT (Wortman, Duket, Seifert, Hann, and Chubb, 1978); and MicroSAINT (MicroAnalysis, 1987). These are general simulation environments and do not, in general, contain embedded operator models. However, several packages are available which combine a task network simulation facility with basic models of the human information processor, for the specific purpose of operator workload and performance prediction (CREWCUT, HARDMAN III, PERCNET, SWAS, TAWL/TOSS, WAM etc.). The Defence and Civil Institute of Environmental Medicine, in collaboration with the Canadian Marconi Company, Kanata, Ontario, have developed a software package for front-end systems analysis that includes a task network simulation facility containing an embedded operator model which is currently based on the VACP model of Aldrich, Craddock, and McCracken (1984). The Systems Operator Loading Evaluation (SOLE) software consists of a software shell and relational data base coupled with SAINT Version 5 as the environment for network simulation (CMC, 1994).

This report outlines the implementation, for use in task network simulation, of a theoretical framework for a new model of the human information processor. The development of the Information Processing (IP) Model is described in detail elsewhere (Hendy, 1994a; Hendy, Liao, and Milgram, 1994). This report deals only with those aspects that are necessary to take the ideas of the IP Model and adapt them for direct application to task network simulation. For more specific details on the theoretical underpinnings of the IP Model, the reader is referred to the previously cited documentation. The material contained in this report provides the bridge between the conceptual descriptions of the IP Model reported earlier, and the software requirements necessary to put that concept into practice. As part of this process, many parameters are defined and assigned tentative values so that the model can be run within the task network simulation environment. In addition, aspects are introduced that, while not strictly forming part of the IP Model itself, are appropriately discussed within the context of integrating a human model within task network simulation. For example, the Sections on *Performance Shaping Factors*, *Task Description Fields* and *Report* generation.

In devising this implementation, many assumptions were made that are beyond the scope of the study reported in Hendy, *et al.* (1994). Hence, further validation is required to verify the IP Model within the context of this implementation. It is intended that SOLE will provide the testbed for implementing and validating the approach described in this Report. As the specific purpose of the report is to describe the algorithmic and data base requirements for implementing the IP model in SOLE, some of the material is specific to that environment. However, much of the material is of a more general nature and could be adapted to other software environments.

THE IP MODEL

The implementation described in the following sections is based on a capacity limited model of the human information processor (Hendy, 1994a; Hendy, *et al.*, 1994). The IP Model is represented schematically in Figure 1.

Each task in any human activity is assumed to involve a certain amount of uncertainty (say B_r bits of information) to be resolved. This is shown in Figure 1 as the *Task Load*. Assuming a processing rate of C bit s^{-1}, this leads to a decision time of T_r seconds. It is argued that all behaviour is time constrained in some fashion so that when T_r is compared to the time allowable for initiating a response T_a, *time pressure* is generated. The operator reacting to the imposed time pressure may attempt to adapt this information processing loop by three methods: (1) by reducing the amount of information to be processed by choosing simpler strategies, dropping feedback etc.; (2) by increasing processing rate or channel capacity; or (3) by increasing the time available for making a response, say by allowing error to accumulate for a longer period of time before attempting to null to zero. It is asserted that methods 1 and 3 are the most likely adaptations, with method 2 restricted to small variations in processing rates in response to changing physiological and psychological states such as arousal, activation, fatigue, motivation, anxiety etc.

Figure 1 attempts to bring together many of the concepts and constructs relevant to operator performance and workload, and show the relationships between them. For example: *Task Difficulty* is associated with *Task Load* in bits of information; *Capacity* with *Processing Rate*; *Effort* with the process of adaptation which both drives physiological and psychological states and is in turn influenced by them. Yet, the basic idea of the IP model is really very simple and is represented by the two equations, shown in Figure 1, that trace the conversion of *Task Load* into *Time Pressure*. Figure 1 should be read according to the following semantic model. For example, one path in Figure 1 (starting with *Time Pressure* and finishing with *Task Load*) can be read as follows "...*Time Pressure* **drives** *Effort* which **adapts** the loop by **decreasing** *Task Difficulty* which is **associated** with *Task Load*."

From the IP Model it can be predicted that (Hendy, 1994a), operator workload and performance depend on the level of *time pressure*. This is determined by the ratio

$$\frac{\text{time required to process information}}{\text{time allowable}}.$$

If the processing rate is assumed to be constant, this ratio can be shown to be

$$\frac{\text{amount of information to be processed}}{\text{time allowable}}.$$

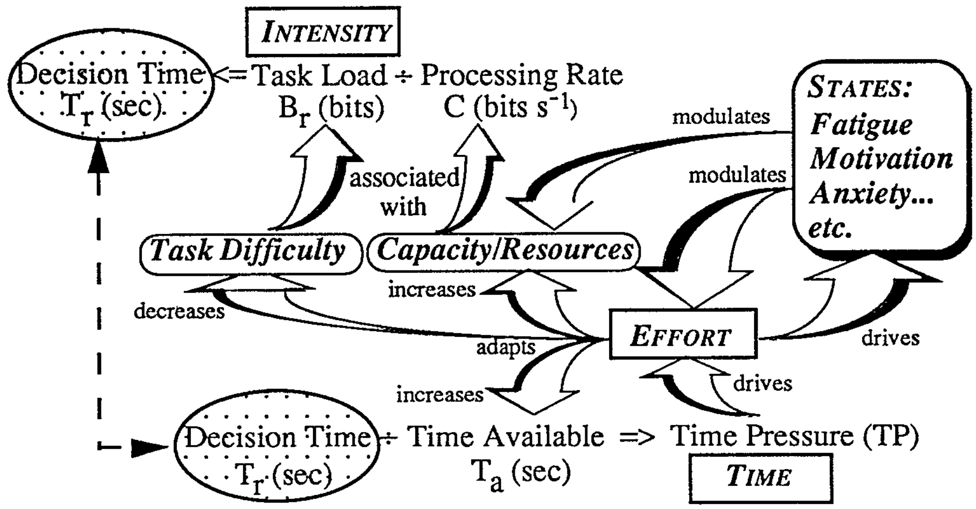


Figure 1. *An information processing (IP) model for the human operator showing the relationships between some of the principal constructs associated with mental workload.*

The assumption embedded in the IP Model is that incoming tasks compete for processing structures that, once allocated to a particular task element, process information in a serial fashion. Therefore, during the time that a particular structure is engaged in processing some element of a given task, it is unavailable to process elements of other tasks. The notion of multiple task interference that is embedded in the IP model, assumes that multiple structures exist for processing and, to the extent that tasks which overlap in the time domain share part or all of a common structure, task interference will be manifested in the form of increased decision times due to serial processing of some or all of the task elements. Further, it is assumed that the components of overlapping tasks that involve a common structure are actually processed serially by time multiplexing.

In the following sections an analytical representation of the IP model is developed for use in task network analysis. This implementation includes the concept of multiple task interference, together with a basic allocation of attention algorithm. The latter is based on task *prioritization* and draws on the notion of *compatibility* as determined by task *interference* effects. The *allocation of attention* algorithm sets limits on the number of tasks that can be performed in concert. Note that the idea of concurrent task performance promoted in the IP Model, consists of a combination of both strictly parallel and serial, time multiplexed, processing of task elements. The data base requirements for implementing the model are also described, together with various recommendations for the use of default values and conditions to allow the rapid building and testing of models.

MULTIPLE TASK PERFORMANCE

Suppose that the performance of tasks i and j overlap in the time domain. Then it is assumed that processing two tasks that share a common structure will occur by rapidly time multiplexing within that structure as illustrated in Figure 2 (zero switching time is assumed). In Figure 2, Tasks 1 and 2 are shown to be processed on successive processing intervals. The reaction times of both tasks will be delayed by this form of processing.

Note that:

T_i' is the task completion time of the ith task when performed in isolation, and

T_{ij} is the task completion time of the ith task when performed in combination with the jth task.

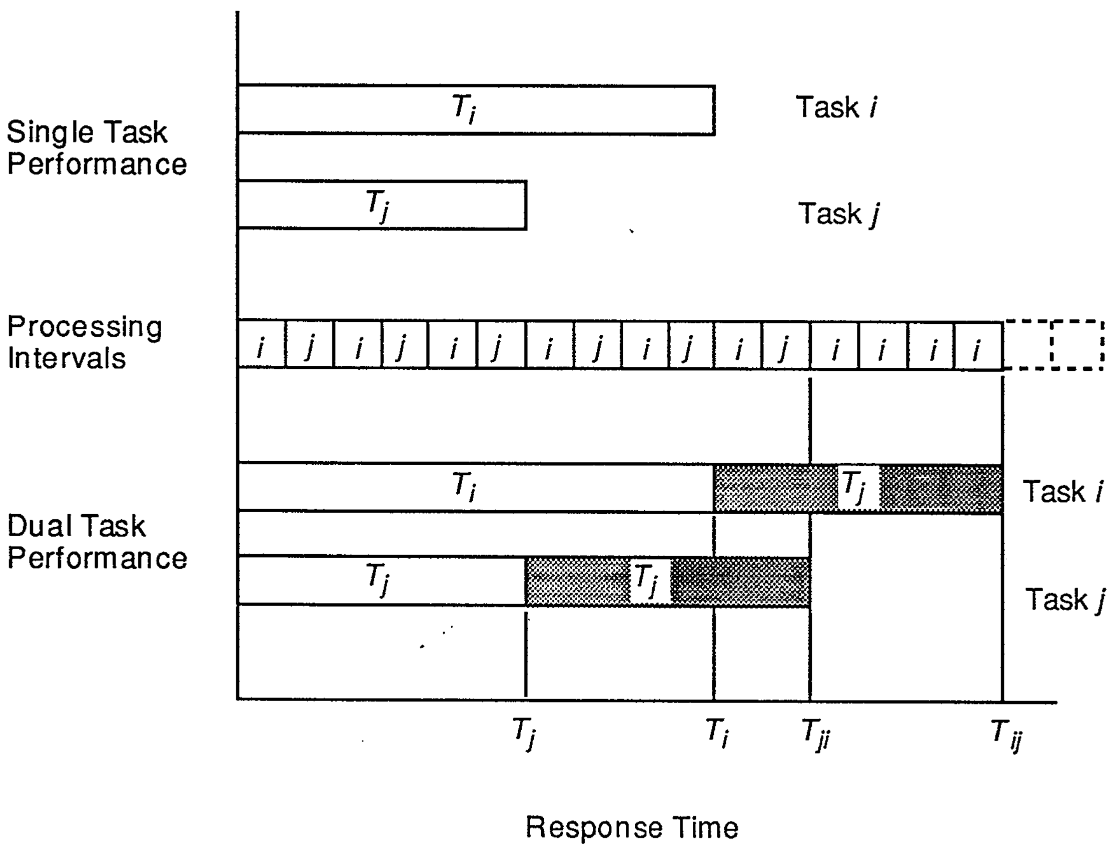


Figure 2. *Time multiplexing in 'concurrent' task processing.*

Suppose that, instead of successively switching from one task to the other, there is a probability associated with the allocation of a processing structure to each task within a given interval (Kinchla, 1980). Assume that in any processing interval, the probability

that the processing structure is assigned to the ith task is p_i. Then on average, over any given time period, a proportion p_i of the processing time is devoted to task i, while a proportion $p_j = (1-p_i)$ is devoted to task j.

In general, suppose tasks i and j do not require the same processing structures for all of their processing time, but share a common structure for a proportion (as defined by the coefficient $\mathbf{c}_{ij}$) of the period of their overlap (see Appendix 1 for complete details of the following derivation). Two cases need to be considered:

> CASE 1 — in which the processing of one of the tasks (assume it to be task j) is entirely embedded within the processing time of the other (task i); and
>
> CASE 2 — in which the tasks partially overlap (the processing time of task i, remaining after task j starts, is entirely embedded within the processing time of task j).

Let

$t_i(\mathrm{s})$ be the starting time of the ith task, and

$t_{ij}(\mathrm{e})$ be the ending time of the ith task when performed in combination with the jth task.

Then for CASE 1, $\left[t_j(\mathrm{s}) - t_i(\mathrm{s})\right] \geq 0$ and $t_{ij}(\mathrm{e}) \geq t_{ji}(\mathrm{e})$, and assuming that the requirement to share common structures is distributed evenly throughout the period of overlap, it can be shown that

$$T_{ij} = T_i + \frac{\mathbf{c}_{ij}(1-p_i)}{1-p_i\mathbf{c}_{ij}} T_j,$$

$$T_{ji} = \frac{T_j}{1-p_i\mathbf{c}_{ij}}, \text{ and}$$

$$T_i - T_j\left[\frac{1-\mathbf{c}_{ij}(1-p_i)}{1-p_i\mathbf{c}_{ij}}\right] - \left\{t_j(\mathrm{s}) - t_i(\mathrm{s})\right\} \geq 0.$$

Similarly, for CASE 2, $\left[t_j(\mathrm{s}) - t_i(\mathrm{s})\right] \geq 0$, $t_{ij}(\mathrm{e}) < t_{ji}(\mathrm{e})$, and

$$T_{ij} = \frac{\left[p_j\mathbf{c}_{ij}\left\{t_j(\mathrm{s}) - t_i(\mathrm{s})\right\} + T_i\right]}{1-p_j\mathbf{c}_{ij}},$$

$$T_{ji} = \frac{(1-p_j\mathbf{c}_{ij})T_j + \mathbf{c}_{ij}(1-p_j)T_i - \mathbf{c}_{ij}(1-p_j)\left\{t_j(\mathrm{s}) - t_i(\mathrm{s})\right\}}{1-p_j\mathbf{c}_{ij}}, \text{ and}$$

$$T_i - T_j\left[\frac{1-\mathbf{c}_{ij}(1-p_i)}{1-p_i\mathbf{c}_{ij}}\right] - \left\{t_j(\mathrm{s}) - t_i(\mathrm{s})\right\} < 0.$$

In these two sets of expressions, the inequality classifies the situation according to case. If the proportion of the time devoted to processing each task is determined by task priorities P_i and P_j, then,

$$p_i = \frac{P_i}{P_i + P_j}.$$

To allow for task resumption after an interruption it is necessary to keep a running total of the amount of actual processing time devoted to each task. At any time $t_j(\mathrm{s}) \le t \le t_{ji}(\mathrm{e})$, the amount of processing time devoted to task i, since task j commenced, is

$$\Delta T_{ij} = p_i \mathbf{c}_{ij}\{t - t_j(\mathrm{s})\} + (1 - \mathbf{c}_{ij})\{t - t_j(\mathrm{s})\}$$

and to task j

$$\Delta T_{ji} = (1 - p_i)\mathbf{c}_{ij}\{t - t_j(\mathrm{s})\} + (1 - \mathbf{c}_{ij})\{t - t_j(\mathrm{s})\}.$$

Therefore, at time t the amount of processing time remaining on each task is

$$T_i - \sum_j \Delta T_{ij} \text{ for task } i, \text{ and}$$

$$T_j - \sum_i \Delta T_{ji} \text{ for task } j.$$

These relationships for the task completion times under the two cases produce Performance Operating Characteristics (POCs) for the IP Model, as shown in Figure 3. The following assumptions were made in generating these curves;

$$T_i = T_j = 1\mathrm{sec}; \text{ and}$$
$$t_i(\mathrm{s}) = t_j(\mathrm{s}).$$

The five POCs of Figure 3 are for values of $\mathbf{c}_{ij} \in \{0.05, 0.25, 0.5, 0.75, 0.95\}$ and priorities in the range of 0.01 to 0.99.

It is of interest to compare the POCs of Figure 3 with those derived from another time multiplexed model, namely, Shaw's search model (Kinchla, 1980). To make these comparisons, consider the curve for $\mathbf{c}_{ij} = 0.95$ in Figure 3. Shaw's model (see Figure 11.9 in Kinchla, 1980) shows response latencies approaching ∞ as either $p_i \to 1$ or $p_i \to 0$, whereas the IP Model asymptotes to values which are consistent with the prediction of serial processing. Both models provide equivalent estimations for $\mathbf{c}_{ij} = 0.5$. Overall the predictions of the IP Model are more reasonable, as one would expect the response latency of the lower priority task $(p \to 0)$ to approach $(T_i + T_j)$ rather than ∞.

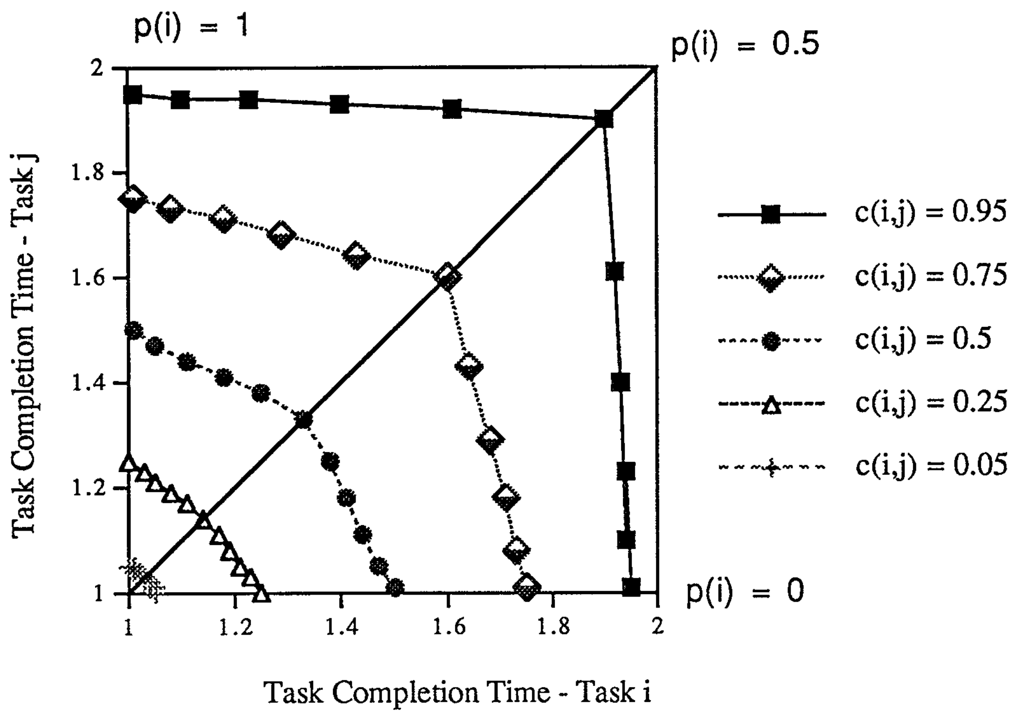


Figure 3. *Performance Operating Characteristics for the IP Model, for various values of task interference* (c_{ij}) *and priority given to processing task* i ($p(i)$).

For multiple task performance, it is assumed that two types of interference can occur, namely, *structural* interference and *resource limited* interference. The term *structural* interference is used, quite specifically in this context, to describe interference effects that are due to limitations such as:

- the inability to focus foveally at different images, widely separated in visual angle;
- those problems associated with operating different controls with the same hand or limb;
- those problems associated with attending to the content of two auditory messages at the same time; and
- the inability to speak two messages at the same time etc.

Structural limitations have nothing to do with the processing structures involved, at least at the higher levels of processing. They are assumed to be associated with the input and output stages, rather than cognition. Structural interference is assumed to be *all or nothing*, that is c_{ij} = 1 or 0. Matrices of interference coefficients for vision, audition, manual and kinesthetic domains are shown in Tables 1 to 3. Default values are shown for completeness. These are the values that will be assigned when tasks are initialized. Note that these assignments have been made arbitrarily and no claims are made for their validity.

A *home* area will be defined for vision, which is assumed to be the resting position for the eyes in the absence of specific operator initiated eye movements. At the end of each task

that involves a visual component, the direction of gaze will be returned to this point. This allows the location of externally initiated visual stimuli to effect the outcome of the task network simulation, say, through the probability of detecting and responding to a new visual stimulus occurring in an area other than the *home* area.

Within the cognitive domain it is assumed that resource limited performance stems from the competition for common processing structures. In this domain, the degree of interference is graded, with coefficients taking values in the range 0 to 1 (see Table 4). At some level of interference it seems reasonable to assume that tasks will be performed in a strictly serial fashion rather than be time multiplexed. The difference between serial and interleaved performance will be seen in the position of tasks on the simulated timeline. For interleaved performance, task start times remain unchanged by interference effects, but the completion time of tasks are delayed. With strictly serial performance, one task is postponed until the other has been completed; therefore, there will be changes in the task start and stop times. The times required for the tasks to be processed, however, are not modified in this case. Tentatively, this critical value is set at $\mathbf{c}_{ij} \geq 0.7$. Note that the only interference coefficients that exceed 0.5 in Table 4, are those for which $i = j$ or involve the default category *no allocation*. However, as these are arbitrary assignments, the situation may change in the future.

TABLE 1

Task interference coefficients ($\mathbf{c}_{ij}$) for a human information processing model — visual input.

Channel	*Mode*	*Interference*
Visual	Input	Structural ($\mathbf{c}_{ij}$ = 1 or 0)

	Angular subtence (degrees)		
Categories	$\varnothing \leq 2°$	$2° < \varnothing < 30°$	$\dagger\varnothing \leq 30°$
1. Central-Central	**0.0**	**1.0**	**1.0**
2. Central-Peripheral	**0.0**	**0.0**	**1.0**
3. Peripheral-Peripheral	**0.0**	**0.0**	**1.0**
4. No allocation (default) with any Central or Peripheral task	**1.0**	**na**	**na**

† *These values are for operator initiated eye movements. Note that an externally initiated visual signal, occurring outside a certain angle (say 30°) of visual arc, will not be detected — or will be detected with a certain probability — and therefore may not be 'serviced'. Tasks not allocated will be serviced.*

TABLE 2

Task interference coefficients ($\mathbf{c}_{ij}$) for a human information processing model — vocal and auditory domains

Channel	*Mode*	*Interference*
Auditory	Input	Structural ($\mathbf{c}_{ij}$ = 1 or 0)
Vocal	Output	Structural ($\mathbf{c}_{ij}$ = 1 or 0)

Categories	1	2	3	4	5	6
1. Tone or simple auditory signal	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**1.0**
2. Speech input (*incidental to primary task*)	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**1.0**
3. Auditory Pattern	**0.0**	**0.0**	**1.0**	**1.0**	**1.0**	**1.0**
4. Speech input (*attended to, relevant to primary task*)	**0.0**	**0.0**	**1.0**	**1.0**	**1.0**	**1.0**
5. Voice output	**0.0**	**0.0**	**1.0**	**1.0**	**1.0**	**1.0**
6. No Allocation (default)	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**

TABLE 3

Task interference coefficients ($\mathbf{c}_{ij}$) for a human information processing model — psychomotor and kinesthetic domains.

Channel	*Mode*	*Interference*
Tactile	Input	Structural ($\mathbf{c}_{ij}$ = 1 or 0)
Manual	Output	Structural ($\mathbf{c}_{ij}$ = 1 or 0)

Categories	*Categories*
Left Hand and Right Hand	Left Leg and Right Leg
Whole hand	*Whole leg*
Digit 1	*Foot*
Digit 2	
Digit 3	No Allocation (default)
Digit 4	
Digit 5	

Categories	Yes	No
Same arm, leg, foot, finger, at least one task is not allocated etc.	**1.0**	**0.0**

TABLE 4

Task interference coefficients ($\mathbf{c}_{ij}$) for a human information processing model — cognitive processing.

Channel	*Mode*	*Interference*
Cognitive	Central	Resource Limited (see interference matrix — note that structural interference will take precedence over resource limitations, and the maximum value of $\mathbf{c}_{ij}$, over all domains, will be used to determine the outcome)

Categories	1	2	3	4	5
1. Automatized, highly learned	0.0	0.0	0.0	0.0	0.0
2. Verbal encoding, decoding, speech production	0.0	1.0	0.2	0.5	1.0
3. Spatial encoding decoding, pattern recognition	0.0	0.2	1.0	0.3	1.0
4. Memorization/recall, calculation, estimation, deduction, reasoning	0.0	0.5	0.3	1.0	1.0
5. No allocation	0.0	1.0	1.0	1.0	1.0

A miscellaneous channel (Table 5) is included to account for effects that are not covered adequately by the modalities of Tables 1 to 4. For example, one may wish to model team activities rather than individual operator tasks, redefine the categories for one of the domains, or introduce a new modality such as aided vision. The miscellaneous channel provides some flexibility for accommodating additions such as these to the network of tasks.

It should be possible to transform and combine the $\mathbf{c}_{ij}$ values for the miscellaneous categories of Table 5 with any or all of the $\mathbf{c}_{ij}$ values from Tables 1 to 4, together with the states of tasks i and j, by standard mathematical and logical relationships. Some examples are:

- if category(vision).EQ.3
  then
  $\mathbf{c}_{ij} = \mathbf{c}_{ij}$ (cognitive) + 0.1
  else
  $\mathbf{c}_{ij} = \mathbf{c}_{ij}$ (cognitive)

- $\mathbf{c}_{ij} = \mathbf{c}_{ij}$ (cognitive) + $\mathbf{c}_{ij}$ (misc.) etc.

The following operations should be supported as a minimum requirement:

Mathematical	: +, -, ×, ÷;
Transformations	: logarithmic, trigonometric, polynomial; and
Logical	: if—then—else, =, ≠, max, min.

The value of $\mathbf{c}_{ij}$ used to determine the increase in task completion times, will be the maximum value of the coefficients obtained from all active domains represented by Tables 1 to 4, and the value of $\mathbf{c}_{ij}$ obtained from these transformations. Obviously structural interference ($\mathbf{c}_{ij}$ = 1), when present, will dominate.

TABLE 5

Task interference coefficients ($\mathbf{c}_{ij}$) for a miscellaneous processing domain.

Channel	*Mode*	*Interference*
Miscellaneous		Structural or Resource limited.

Categories	1	2	3	4	5	...	*n*
1. Category 1	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	...	**1.0**
2. Category 2	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	...	**1.0**
3. Category 3	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	...	**1.0**
4. Category 4	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	...	**1.0**
5. Category 5	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	...	**1.0**
...	...	...	...	...	...	...	...
n. Category *n*	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	...	**1.0**

Finally, it should be possible to enable or disable each of the interference effects independently. If all interference effects are disabled, a default of $\mathbf{c}_{ij}$ = 0 for the working value of this parameter will allow completely parallel processing independent of the state of the allocation of attention module. Note that the default values of Tables 1 to 5, when interference effects are enabled, will force strictly serial processing.

ALLOCATION OF ATTENTION

In general, a task network could be said to simulate the demand placed on the operator by the system, rather than the task load actually serviced by the operator. A task network can have many parallel branches which leads to the generation of multiple concurrent task demands. In many cases these demands clearly exceed human capabilities to respond (e.g., CMC, 1992; Glenn, Cohen, Wherry Jr., and Carmody, 1994). The purpose of an *allocation of attention* module is to schedule the tasks to be performed, either serially or concurrently, at any point in time. The allocation of attention module will determine whether a task is performed on demand, interrupted, resumed, postponed, or shed. The allocation of attention algorithm should provide a fair representation of human task selection strategies under competing system demands. The SOLE software incorporates a rudimentary allocation of attention algorithm called the Task Conflict Alleviation Technique (TCAT), however, it is driven by a set of rules (CMC, 1994) that have a fundamentally different basis than the structure of the IP Model.

The first assumption to be made, and perhaps the most fundamental, is that operators will service no more than 2 tasks concurrently for which $\mathbf{c}_{ij} > 0$ (Hendy, 1994a). While the literature on dual task performance is abundant, information on multiple (more than 2) task performance is rare or non-existent. While the restriction to dual tasking probably provides a conservative prediction, overt multiple task performances appear to be rare in operational systems (Shaffer, Hendy, and White, 1988). Dual tasking is limited to tasks that require higher level processing, say at the level of Rasmussen's rule-based and knowledge-based activities (Rasmussen, 1983). There is no limit set on the number of purely skill-based activities (for which it is assumed that the cognitive component $\mathbf{c}_{ij} = 0$) that can be performed in concert provided there are no structural interference limitations.

When the task network generates a new demand, a set of rules will govern the scheduling of tasks: When new tasks arrive or an ongoing task finishes, a temporary queue will be generated containing all tasks currently running, together with any new tasks and tasks awaiting attention (these are retrieved from a short term *memory* queue). A repeating task (i.e., any task that is programmed to repeat at regular intervals — see discussion on 'continuous' tasks) will not be added to the temporary queue if its predecessor remains present in the short term queue. Neither will it be transferred to the short term memory queue following the current task scheduling. Task scheduling will be in accordance with the following rules based on *priority*, *interuptability/resumability* and *sheddability*. Note that, in general, *priority* may be time or state dependent (e.g., the priority of a display may increase with time since last glance, the priority of a task may change due to the occurrence of some predisposing condition). Short term memory queue size will be tracked. This queue will be flushed on a *first opportunity* basis (i.e., as soon as an ongoing task finishes, the queue will be examined to see if there are any tasks that can be started).

> *Rule 1.* Active tasks, which are deemed to be *interuptable*, may be halted if less than C_{crit} complete (tentatively $C_{crit} = 70\%$) — an interrupted task may be resumed or restarted later. An *uninteruptable* task, once started, must run to completion. A task which is not resumable will retain its place in the queue and be restarted if possible. Task interruptions will be logged.

Rule 2. Tasks for which $\mathbf{c}_{ij} = 0$ will be started on demand.

Rule 3. Tasks for which $0 < \mathbf{c}_{ij} \leq 1.0$, including interrupted tasks, will be serviced in order of *priority*. All task postponements will be logged. Task(s) of the highest *priority* value(s) will be serviced first.

Rule 4. If several tasks have the same priority they will be scheduled according to the following hierarchy:

- in order of their originally scheduled start time (including non-resumable tasks that are restarting), and independent of the number of interruptions;
- in order inverse of their processing time remaining;
- according to least interference; and
- a random selection will be made.

Rule 5. If a task is delayed it will be allocated to a less loaded channel if possible.

Rule 6. If a task is deemed *sheddable*, it will be permanently removed from the short term memory task queue (and logged as such) after n (tentatively, $n = 3$) unsuccessful attempts to start or reschedule it. Repeating tasks that are shed due to an unprocessed predecessor, will be similarly logged.

Rule 7. The short term queue will be limited to m items (tentatively $m = 3$). On transfer from the temporary queue, tasks will be shed from the bottom of the priority list, *sheddable* tasks in order of the number of scheduling attempts, to meet this limit. Tasks shed will be logged. Tasks partially serviced when shed will have the completed processing time logged (% complete).

Several possibilities for modelling human performance flow from this scheduling algorithm. For example, the probability of a successful outcome for some tasks may decrease if interrupted or delayed, or a task may be dropped from the queue (forgotten) if not serviced within a certain time period — most likely the probability that an item is forgotten would increase with time or the nature of other tasks in the queue (see the discussion on memory in Card, Moran, and Newell, 1983). The specific implementation of memory effects in a task network simulation are likely to be application specific, hence a detailed discussion is beyond the scope of this report. However, the requirement to allow for such representations is flagged so that software development provides the necessary tools for implementing these concepts. At this stage, it will be sufficient to provide access to certain system variables such as:

- number of interruptions;
- list of active tasks (running and in the queue);
- % processing completed;
- time since scheduled start time; and
- categories (from Tables 1 to 5) of all tasks in the queue.

There are likely to be pairs of tasks that although predicted to be structurally or resource limited (tentatively for values of $\mathbf{c}_{ij} \geq 0.7$) may be compatible in certain combinations (e.g., controlling aircraft pitch and roll with a joystick or control wheel involves the same

hand but is a compatible combination). Allowance will be made for these exceptions in the task description data entry form. Note that if all task interference effects are disabled the allocation of attention module becomes superfluous and therefore should be disabled.

MICRO MODELS

The representation of task completion times and error rates in task network simulation is crucial. They can be estimated from human performance data, and a variety of micro models of the human operator gleaned from the literature. Following is a summary of some of the information available in CREWCUT (Little, Dahl, Plott, Wickens, Powers, Tillman, Davilla, and Hutchins, 1993), which is useful for estimating response times and error rates in a variety of situations when alternative empirical data are unavailable.

<table>
<tr><td>1.</td><td>Eye Movement Time — target located in eye field</td><td>100ms (travel time only)</td></tr>
<tr><td>2.</td><td>Head Movement Time— target located in head field</td><td>200ms (travel time only)</td></tr>
<tr><td>3.</td><td>Listening— accuracy is a function of signal to noise ratio (dB) and noise interruption frequency as follows:
<table>
<tr><th>Error Rate†</th><th>S/N Ratio (db)</th><th>Noise Interruption Frequency (Hz)</th></tr>
<tr><td>0.10</td><td>9</td><td>1</td></tr>
<tr><td>0.07</td><td>9</td><td>10</td></tr>
<tr><td>0.10</td><td>9</td><td>100</td></tr>
<tr><td>0.20</td><td>0</td><td>1</td></tr>
<tr><td>0.12</td><td>0</td><td>10</td></tr>
<tr><td>0.25</td><td>0</td><td>100</td></tr>
<tr><td>0.35</td><td>-9</td><td>1</td></tr>
<tr><td>0.25</td><td>-9</td><td>10</td></tr>
<tr><td>0.65</td><td>-9</td><td>100</td></tr>
<tr><td>0.42</td><td>-18</td><td>1</td></tr>
<tr><td>0.28</td><td>-18</td><td>10</td></tr>
<tr><td>0.96</td><td>-18</td><td>100</td></tr>
</table>
† Assumes a 50 word vocabulary. Error rate would increase with a larger vocabulary and decrease with a smaller vocabulary.</td><td>2.4 words per second</td></tr>
<tr><td>4.</td><td>Eye Fixation Time</td><td>100 - 500ms</td></tr>
<tr><td>5.</td><td>Search
N = number of fixations
T_m = movement time
T_f = fixation time</td><td>Time = $(T_m + T_f)N$</td></tr>
</table>

6.	**Reading Rate**— *assumes 5 letter words, 2.5 words or 13 characters per phrase*	52 words per minute (5 saccades per word) 261 words per minute (1 saccades per word) 652 words per minute (1 saccades per phrase)
7.	**Hand Movement** — *Welford's variant of Fitt's Law.* T_H = movement time D = distance between targets S = size of target I_H = slope constant (0.1 second per bit)	$T_H = I_H \log_2(D/S + 0.5)$
8.	**Operate Pushbutton or Toggle**	400ms
9.	**Operate Rotary Dial**	730ms
10.	**Cursor Movement with Trackball** — *from Fitt's Law* T_T = cursor positioning time D = cursor distance to be moved S = display symbol width I_T = slope constant (0.1 second per bit)	$T_T = I_T \log_2(D/S + 0.5)$
11.	**Cursor Movement with Mouse** — *based on Fitt's Law* T_M = cursor positioning time $\mathbf{K}_M$ = 1.03 sec D = cursor distance to be moved S = display symbol width I_M = slope constant (0.06 second per bit)	$T_M = \mathbf{K}_M + I_M \log_2(D/S + 0.5)$
12.	**Cursor Movement with Joystick** — *based on Fitt's Law* T_J = cursor positioning time $\mathbf{K}_J$ = depends on D as follows D — K_J 160mm — 1.68sec 80mm — 1.44sec 40mm — 1.26sec 20mm — 1.12sec 10mm — 1.05sec D = cursor distance to be moved S = display symbol width I_M = slope constant (0.1 second per bit)	$T_J = \mathbf{K}_J + I_J \log_2(D/S + 0.5)$

13.	**Cursor Movement with Step Keys** — *based on Fitt's Law* T_S = cursor positioning time K_S = 0.98 sec D_x = cursor distance to be moved horizontally S_x = size of the horizontal step D_y = cursor distance to be moved vertically S_y = size of the vertical step I_S = slope constant (0.074 second per bit)	$T_S = K_S + I_S \log_2(D_x/S_x + D_y/S_y)$
14.	**Cursor Movement with Text Keys** — *based on Fitt's Law* T_{text} = cursor positioning time K_{text} = 0.66 seconds κ_{text} = keystroke rate (default is 0.209 seconds per keystroke) N_{min} = minimum number of keystrokes	$T_{text} = K_{text} + \kappa_{text} N_{min}$
15.	**Single Finger Keying Rate**	0.140 (0.06 to 0.20) seconds
16.	**Typing Rate**	0.209 seconds per keystroke
17.	**Walking Rate**	0.62 seconds per meter
18	**Speech Production**	2.4 words per second for a small vocabulary 3.4 words per second for a large vocabulary
19.	**Cycle Time for the Perceptual Process of the Model Human Processor** (see Card, *et al.*, 1983)	$\tau_p = 100ms$
20.	**Cycle Time for the Cognitive Process of the Model Human Processor of the Model Human Processor** — one cycle per attribute (see Card, *et al.*, 1983)	$\tau_c = 70ms$
21.	**Cycle Time for the Motor Process of the Model Human Processor** (see Card, *et al.*, 1983)	$\tau_m = 70ms$
22.	**Simple Reaction Times** *On/Off response* *Physical Match* *Name Match* *Class Match*	$\tau_p + \tau_c + \tau_m = 240ms$ $\tau_p + 2\tau_c + \tau_m = 310ms$ $\tau_p + 3\tau_c + \tau_m = 380ms$ $\tau_p + 4\tau_c + \tau_m = 450ms$

23.	**Choice Reaction Time** — Hick's Law *RT* = reaction time *n* = number of alternatives **K** = 150ms (or 240ms from item 22 above)	$RT = K\log_2(n + 1)$
24.	**Mental Rotation** *RT* = reaction time *R* = amount of rotation from perceived to visualized view (degrees)	$RT = 1 + R/50$ seconds
25.	**Prioritization** — e.g., number of targets *RT* = reaction time *n* = number of targets	$RT = 0.31(n(n\text{-}1)/2)$
26.	**Terrain Association** — *a two stage process* *Stage 1 — reduce the size of the area of uncertainty (performed once every time a completely new view is encountered)* *Stage 2 — pinpoint own location (4 to 7 matching attempts are required; usually interspersed with other activities)*	5 seconds 2 seconds per terrain matching attempt

PERFORMANCE SHAPING FACTORS

Task network simulation has the potential to be sensitive to a number of performance shaping influences such as: manpower (crew size), personnel (aptitude, command, experience) and training (knowledge, skills) issues; fatigue and other physiological stressors; allocation to an alternative processing channel (another sensory channel or operator); operator adaptation to high information processing loads; and various psychological stressors. The network properties that are available to implement this potential are (Hendy, Kobierski, and Youngson, 1992):

- individual task inventory;
- task sequence, including branching due to conditional or probabilistic task outcomes (e.g., resulting from a changed probability of successful task completion); and
- task completion time.

The IP Model (Hendy, 1994a; Hendy, *et al.*, 1994) carries with it the assumption that operators adapt their processing loop to increasing time pressure by resorting to strategies that involve less information to be processed or provide greater time for response. This results in either a decrease in the time required to process the decision or an increase in the time allowable. The original Seigel and Wolf modelling environment (Seigel and Wolf, 1969) incorporated a discontinuous U-shaped modifier, driven by a stress factor which is related to *percent time occupied*, for task completion time. Initially task completion times are reduced as stress increases until a threshold is reached at which point there is a step increase in response times. In terms of the IP Model, Seigel and Wolf's modifier can be interpreted as a reduction in task completion times as less computationally intensive strategies are adopted, followed by increased times as behaviour becomes disorganized due to task shedding and increased error rates, or response times are prolonged due to attempts to service conflicting tasks (i.e., those that share common structures). Initially, a simple implementation should be adopted in which it will be possible to modify task completion times by a simple linear factor, varying from 0 to $y\%$, as the time pressure variable (see following Section) increases from 0 to 100%.

As time pressure increases, task shedding will become more prevalent. In the IP Model, error is assumed to depend on the amount of relevant information presented but left unprocessed. At the level of task description used in most network simulations (rarely lower than the button pushing stage) tasks are usually considered completed or not. However, either the time to complete and/or the probability of successful completion of some tasks may depend on the successful completion of various predecessor tasks. For example, the time to respond to an emergency may depend on the extent to which systems states have been monitored recently, a radio can not be set unless the message giving the channel setting was attended to etc. In the initial implementation of the IP Model, it is required that the outcome of a task should be conditional on the successful completion of predecessor tasks, and task completion times can be modified according to the proportion of predecessor tasks (including repeating or continuous tasks) completed within a given time window. The modifying expressions will be assembled from standard algebraic and logical relationships.

The IP Model posits that different levels of experience, knowledge etc. result in different choices of strategies for processing, which effects both the total amount of information to

be processed (hence the processing time) and the processing structure involved (from automatized, perhaps dedicated, structures to algorithmic problem solving involving calculation, recall, use of working memory etc.). These effects could be modelled within a task network environment by changing the task completion times by an appropriate factor and changing the cognitive category of some tasks from Category 1 (Automatized) to Categories 3 or 4 (Spatial encoding decoding, pattern recognition; or Memorization, recall, calculation, estimation, deduction, reasoning) and from Category 3 to Category 4. Similar schemes have been implemented in CREWCUT and HARDMAN III.

In CREWCUT, simple fixed multipliers are used to increase or decrease task completion times to account for 3 levels each of *aptitude* (high, average, and low) and *experience* (experienced, average experience, and inexperienced). CREWCUT introduces a ±15% correction for *aptitude*, and a ±10% correction for *experience.* It is not clear if individual factors are combined, either additively or multiplicitively, in CREWCUT. In HARDMAN III, more complex relationships (at least at the level of linear regression equations) are used. To account for fatigue, CREWCUT multiplies task completion times by a factor which is a function of the time on task, as follows;

$$\text{correction factor} = \frac{1}{\left[0.25\left(0.93^{t}\right) + 0.75\right]}$$

where t is the number of hours of steady task performance. For those tasks that have a probabilistic outcome, error rate may be effected also.

When a task is allocated *a priori* to a preferred processing channel (e.g., a limb or digit) it will be assumed that a time penalty may occur if it is re-allocated, at run time, to a non-preferred channel (e.g., performed by the left rather than the right hand). Initially a fixed penalty of 10% will be assumed. Later more elaborate rules may be developed that are dependent on the target for re-allocation.

Various physiological and psychological states (temperature, noise, vibration, g-stress, chemical agents, drugs, fear, anxiety, motivation, etc.) may work singly or in combination to change task completion times and possibly effect the strategies used to solve problems. This mechanism allows the possibility for linking task models with physiological models (Jensen, 1994). In the simplest form, environmental stressors could be linked to events in the scenario which drives the task network. For example, temperature could increase during a mission, g-stress could be introduced to all tasks associated with an air-to-air engagement etc. It can not be assumed that stressors in combination will act either synergistically or antagonistically. It is likely that complex interactions will occur and the implementation should allow freedom for user defined functions and relationships to be inserted, based on standard mathematical and logical forms.

For the current implementation, it will be assumed that modifying effects are additive (this will be the default condition and possibly reflects a 'worst case' situation). In all cases, these values will be set globally, with the possibility of local modifications for special cases or exceptions. It should also be possible to enable or disable each stressor prior to run time.

MEASURING OPERATOR LOAD

The IP Model posits that operator load depends on the *time pressure*, or the ratio of *time required to process information* to the *time available*. In the context of task network simulation, the time required to process information is given by the task completion time of each activity. An average measure of time pressure can be obtained by computing, by a moving average process, the proportion of some fixed time interval which is occupied by information processing activity. In this context, it is noteworthy that the IP Model treats continuous tasks as repetitive (sampled) activities (Hendy, 1994a). The concept of time pressure represents a return to a metric that has a long history in timeline analysis and task network simulation (e.g., see Linton, Plamondon, Dick, Bittner Jr., and Christ, 1989). The major difference between the current implementation and past usage, with the exception of Wingert's function interlace method (Wingert, 1973), is that with the current implementation performance is sensitive to the presence of multiple concurrent tasks.

Note that in this method of assessing operator load it is necessary to make the distinction between scenario-driven and task-driven network behaviour. Normally, task networks are run under simulation such that as soon as one task finishes the next task starts (i.e., the time behaviour of the network is task driven). Under these circumstances the network should run from the first task to the last task without break. Total simulated mission time is then the sum of the individual task completion times. Alternatively, a scenario driven network would have discrete tasks, or at least groups of tasks, locked to events on an externally prescribed timeline. Hence, there is likely to be periods when no operator activity is called for in the simulation.

For an entirely task driven network the proportion of time occupied with information processing activity is 100% for the duration of the simulation; hence the *time pressure* metric has no sensitivity to changes in the network composition (apart from. possible variations in the total simulated mission time). Traditionally, *continuous* tasks (such as flight control) have led to this type of problem with previous time-based metrics of operator load. The implementation described in this report avoids such problems with continuous tasks, by modelling them as repeating discrete tasks. However the possibility still exists that when there will be many parallel tasks competing for attention, the scheduling of discrete tasks will cause the simulated operator to be occupied for 100% of the mission time. When this represents operator involvement with tasks critical for system operation, this is a fair representation of workload. However, if the load is made up of many tasks which can be shed without great consequence, this is not a fair representation of imposed load.

Typically the task queue will be loaded with monitoring and other non-essential tasks, that while adding to the general situation awareness and knowledge of system state, can be shed without a catastrophic breakdown in system performance. Tasks designated as *sheddable* should not be included in the calculation of the *time pressure* metric. Such tasks may be regarded as *discretionary*, that is the operator can decide to service them or not depending on the status of the *non-discretionary* load. To allow for tasks that change in priority over time (e.g., some system monitoring tasks) *sheddable* tasks that reach the highest *priority* value should be added to the base load and included in the workload calculation.

Certainly, it can be argued that for best operator situation awareness one should be occupied 100% of the time, gathering as much information about the system state as one can. In a life threatening situation any other strategy is inadvisable. However, from the systems engineer's point of view, it is the base load of non-discretionary activities that carries the highest priority for design. These can be considered the imposed or externally paced load. For comparison a loading analysis with all tasks included should be performed as well.

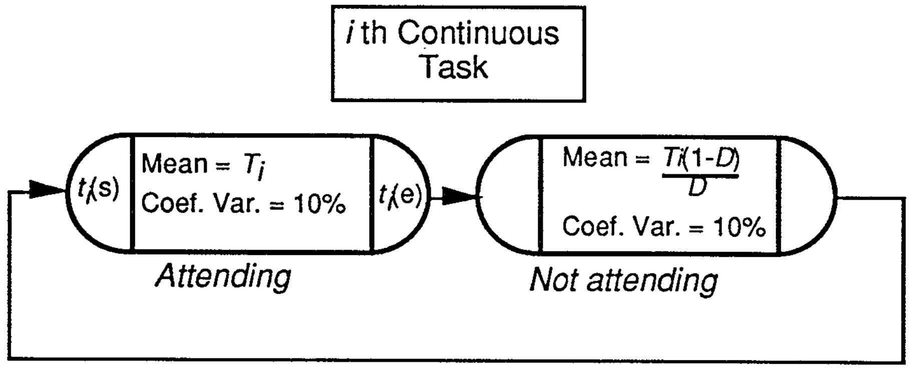
The choice of window size for the moving average depends on the context (see Table 5 for some values used by other authors). Obviously the window should be longer than most task completion times (otherwise predicted time pressure will tend to cycle between 0 and 100%) which in turn depends on the grain of the task analysis. Generally, a task breakdown to the button pressing level will involve tasks of shorter duration than a decomposition that ends at the level of "...select radio frequency." A window that is four times as long as the mean duration of the longest task, is perhaps a reasonable starting point. This value should be selected by the analyst (say, 1 minute, with a 1 second resolution as default values) as one of the simulation parameters, and/or the process could be automated by a parser that detected the longest mean task completion time in the data base and set the window at *n* times this value.

TABLE 5

Window size used in calculating moving averages and the definition of operator overload.

Source	*Moving Average*	*Load Limit*
Linton, Jahns, and Chatelier, 1977	6 seconds	80%
Meister, 1985	5 minutes	75%
Parks and Boucek Jr., 1989	not specified	80%
Malone, Kirkpatrick, and Kopp, 1986	not specified	75%
CMC, 1994	1 minute	none used

As discussed previously, the IP Model treats continuous tasks as repetitive activities. It is assumed that, on average, the proportion of any given time interval devoted to active processing is directly proportional to the rated difficulty of the task. Hence, for continuous tasks, the average duty cycle is set by the difficulty rating D, where $0 < D \leq 1$. The mean time interval devoted to active processing, during each cycle, will be set by the analyst. A default value of 1 second, with an assumed coefficient of variation of 10%, and 50% duty cycle will be used (a Beta, instead of a Normal, distribution could be used, in which case max. and min. values would be set). Hence, the network representation of a continuous task is shown in Figure 4. The starting and ending times for continuous tasks may be set by external factors such as the mission scenario, or from internal network states such as the activation of another task or some parameter taking a particular value or range of values.



t_i(s) start time for continuous task *i*

t_i(e) end time for continuous task *i*

D rated task difficulty

Figure 4. *Network representation of a continuous task.*

While not strictly necessary for purely comparative studies, the specification of a load limit, for defining the point of operator *overload*, tends to be the Holy Grail of workload researchers. Typically, values around 70 to 80% (see Table 5) *time-occupied* are chosen. These values appear to be supported by little more than observations that this marks the point at which load shedding starts. However, empirical evidence is not offered in support of these claims.

For purposes of illustration, suppose that the problem is framed in terms of a single server queuing problem (note that in the current implementation, the single server is sometimes taking customers two at a time). In this situation, some predictions might be made as to what would constitute a point of overload. If tasks are assumed to arrive according to a Poisson process at a constant mean rate λ tasks s^{-1}, and the mean task completion rate also remains constant at a value of μ tasks s^{-1} (assume task completion times are exponentially distributed), then, in a given fixed time interval ∂t

$$
\begin{aligned}
\text{mean time occupied} &= \text{mean number of tasks} \times \text{mean task completion time} \\
&= \partial t \lambda \mu^{-1}, \text{ and} \\
\text{time pressure} &= (\partial t \lambda \mu^{-1}) \partial t^{-1} \\
&= \lambda / \mu .
\end{aligned}
$$

Equating λ with the mean arrival rate of the queuing problem, and μ with the mean service rate (Hillier and Lieberman, 1974), it can be seen that *time pressure* is equivalent to the *utilization factor* ρ of the queuing problem. For a classical single server system, the steady state number of items in the queue is 1 at $\rho = 0.5$, rising to approximately 2 at $\rho = 0.7$, and 4 at $\rho = 0.8$ (Hillier and Lieberman, 1974, Fig 9.6). If the queuing analogy is valid, it seems that a value of *time pressure* around 0.75 is a reasonable limit to set. This would hold the steady state queue size to 2-3 items. It is interesting to note that the values derived from queuing theory seem to be consistent with the limits set from purely empirical studies (e.g., Table 5).

The issue of setting limits on time pressure assumes less importance if overload is redefined in terms of the length and status of the short term 'memory' queue in the allocation of attention module. Of particular interest are occasions of forceful load shedding from this short term storage. This approach strikes directly at what might in fact be the underlying problem of operator overload, and avoids problems associated with the arbitrary selection of parameters for the moving average. In view of the need to distinguish between discretionary and non-discretionary tasks when computing operator load, this shift from a traditional workload paradigm to a concern for tasks serviced versus tasks shed is particularly salient. Basically the analysis shifts from a concern for *workload* to an interest in *errors* (task shed, delayed etc.) and the development of system status knowledge by tracking the proportion of tasks serviced that contribute to situation awareness. Using this approach, task shedding would be tracked and categorized by the type of information involved. Flags should distinguish between tasks that are critical to mission performance, those that contribute to situation awareness, etc. To ease data entry, the *children* of *parent* tasks in the task decomposition, should take the values of the parent task as defaults. Note that in the IP Model, operator error is associated with information unprocessed or shed (Hendy, *et al.*, 1994).

TASK DESCRIPTION FIELDS

In order to implement the mechanisms described in the previous sections there are a number of data fields that should be represented in all task descriptions. These fields involve additions to traditional information such as task completion times, initiating and ending effects etc. The required fields are shown in the following Table:

TABLE 6

Task description fields required for the implementation of the IP Model, together with the default values set at initialization.

Field	*Parameter Value*
Vision	
Enabled	*Yes, No*†
Requirement	*Central, Peripheral, No allocation*†
Visual input§	*Home*†, *Area 1, 2, 3, 4...*
Operator initiated	*Yes*† or *No*
Audition and Speech	
Enabled	*Yes, No*†
Auditory input and voice	*Category* (see Table 2†)
Psychomotor and Kinesthetic	
Enabled	*Yes, No*†
Manual output, Tactile input	*Category* (preferred and alternatives; see Table 3†)
Cognition	
Enabled	*Yes, No*†
Cognitive activity	*Category* (see Table 4†)
Miscellaneous	
Enabled	*Yes, No*†
Number of categories	1 to 15 (7†)
Type of activity	*Category* (see Table 5†)
Transformations	None†, function, relationship etc.
Task Scheduling	
Priority (initial value)	$0 \leq Priority \leq 10$†
Priority modifier	None†, function, relationship etc.
Interruptable	*Yes* or *No*†
Resumable	*Yes* †or *No*
Sheddable	*Yes* or *No*†

TABLE 6 cont.

Field	*Parameter Value*
Memory	
Forgetting	*Yes* or *No*†
Memory decay	*time* to $p = 0.5$ (4 sec.†)
Continuous Tasks	
Continuous task	*Yes* or *No*†
Task difficulty	$0 <$ *Difficulty* ≤ 1 (0.5†)
Attending time per cycle	*seconds* > 0 (0.5 sec.†)
Criticality	
Mission performance	*Yes*† or *No*
Situation awareness (mission)	*Yes*† or *No*
Situation awareness (system)	*Yes* or *No*†
Compatibility	
Compatible tasks	*None*†, *Task pairs*
Performance Shaping Factors	
Adaptation to time stress	*Yes* or *No*†
• Change cognitive category?	*(global logical expression)* or *No*†
Aptitude	*Yes* or *No*†
• Change cognitive category?	*(global logical expression)* or *No*†
Experience	*Yes* or *No*†
• Change cognitive category?	*(global logical expression)* or *No*†
Environmental	*Yes* or *No*†
• Change cognitive category?	*(global logical expression)* or *No*†
Fatigue	*Yes* or *No*†
• Change cognitive category?	*(global logical expression)* or *No*†
Physiological	*Yes* or *No*†
• Change cognitive category?	*global logical expression)* or *No*†
Preferred Limb	*Yes* or *No*†

Notes for Table 6:

§ *Each visual task should be assigned an angular position. By default, all visual activity would be assigned to area 1.*

† *To ease data entry, and facilitate the early testing of task networks during development, the default values shown in this Table, together with the values in Tables 1 to 4, shall be assigned to all tasks.*

To provide flexibility in the task descriptions used in network modelling, any or all of the processing domains, including the miscellaneous domain, shall be capable of being enabled or disabled independently. By disabling features and using default values, it should be possible to start running simulations with a minimum of data entry. A report should be generated listing all tasks where parameters have not been changed from their default values. If a default value is selected as the working value, it should be possible to mark this parameter so that it will be eliminated from future reports.

For the performance shaping factors parameters will be set both globally (e.g., the level of aptitude, experience etc. of the operator; the logical expression that changes the category of cognitive tasks; and the correction factors for each aspect considered) and locally (whether the task is sensitive to the each aspect). Default values will be set at initialization.

REPORTS

The capability of generating the reports shown in Table 7 is required for the SOLE implementation. For implementations in other environments, these may be used as a guide.

TABLE 7

Report generation for the IP Model implementation.

	Item	*Format*
1.	Trace of mean time pressure (± 1 SD) versus normalized mission time.	graphical
2.	Trace of the proportion of times (out of 100 runs) that the short term memory queue size is less than, or equal to, n, (n-1), (n-2), ... versus normalized mission time (say, 1 second resolution) — flagging points where task interruptions, shedding, or delays occur on 50% or more of the runs.	graphical
3.	Table of tasks shed and corresponding tasks performed (for which $\mathbf{c}_{i/j} > 0$) — shown as a proportion of 100 runs, against normalized mission time — categorized by criticality (to mission, to situation awareness etc.), and reason for shedding (number of attempts, priority, predecessor task).	tabular
4.	Table of tasks interrupted and corresponding tasks performed (for which $\mathbf{c}_{ij} > 0$) — shown as the mean number of times out of 100 runs, against normalized mission time — categorized by criticality (to mission, to situation awareness etc.), and reason for interruption (priority, interference, random).	tabular
5.	Table of tasks delayed and corresponding tasks performed (for which $\mathbf{c}_{ij} > 0$) — mean delay from 100 runs, against normalized mission time — categorized by criticality (to mission, to situation awareness etc.), and reason for delay (priority, interference, random, uninteruptable task, task > 70% complete).	tabular
6.	Table listing interference values, for a nominated task pair, in each modality (vision, audition, voice, cognition, manual output) — directly from task data base, independent of the simulation.	tabular
7.	Table showing — at a nominated point (or window) in the normalized mission timeline — the proportion of times (out of 100 runs) each task that was shed, postponed or interrupted, failed in the presence of tasks that were successfully scheduled.	tabular

TABLE 7 cont.

Item	*Format*
8. Table showing display surface attended to (1,2,3,4,...) and kinesthetic or output channel used (limb/digit, voice) versus normalized mission time (later, this might be animated, or linked to rapid prototyping or an anthropometric model).	tabular
9. Table showing all tasks for which default values have not been changed since the data base was created— it should be possible to mark entries, where the default values have been selected as final values, so that these entries will not be listed in future tables.	tabular

Table fields should be separated by tabulation characters so data can be imported into spread sheets such as Lotus 1-2-3® or Excel® or word processing packages such as MicroSoft Word® or WordPerfect®. Graphical outputs should be exportable to word processing packages. A format that allows editing is preferred so that titles can be changed or added and annotations made.

DISCUSSION

This report outlines the implementation of an information processing model for use in task network simulations. This implementation includes a representation of the operator's allocation of attention and human memory, together with a framework for tracking the load on the operator's information processing system. The framework for this implementation is provided by Hendy's (1994a) IP Model. In positing that human information processing load is determined by the ratio of *time required* to *time available*, the IP Model returns to an approach which has many precedents in the history of task network simulation. However, in recent years classical time-based predictions of operator load have largely given way to procedures that attempt to apply the tenets of resource theory to the problem. Current resource-based techniques owe much of their inspiration to the original work of Aldrich, *et al.*, (1984) which in turn has its roots in Wickens' Multiple Resource Theory (Wickens, 1984). These methods largely grew out of an attempt to address deficiencies seen to exist with the traditional time-based approach. These deficiencies include:

- the lack of a theoretical underpinning for the T_r/T_a ratio;
- the inability of time-based methods to discriminate between single and multiple task performance, with the exception of Wingert's (1973) function interlace procedure;
- the insensitivity of time-based methods to the difficulty of continuous tasks; and
- the necessity to treat continuous and discrete tasks separately.

The IP Model, as implemented in this report, answers each of these criticisms. Firstly, the IP Model provides the theoretical framework for claiming time pressure is the primary driver of operator workload, performance and errors. Secondly, the concept of interference in multiple task performance is implemented through the interference coefficients of Tables 1 to 5. Finally, by treating continuous tasks as repetitive discrete activities, the problems associated with the difficulty of continuous tasks and the combination of continuous and discrete tasks are addressed. Note that, in general, the process of validating the IP Model is eased by returning predictions to the time domain.

The introduction of an allocation of attention module allows the focus of the IP Model implementation to shift from solely being that of *workload*, to having a greater emphasis on operator *performance* and *error*. In competitive systems (e.g. most military systems) the operator is likely to be always 100% loaded as survival may depend on constant information processing. The issue then becomes '...loaded with what?' For example, how much time does the operator have to scan the outside scene for targets, how much time is devoted to systems monitoring and navigational updates etc? Hence, the issue for the designer is not so much to reduce the workload imposed by the system, but to maximize the level of performance attainable at a given level of operator information processing capacity. The IP Model implementation attempts to give that insight.

While this report provides the blueprint for building a basic time-based model of the human information processor for task network simulation, the embedded model should not be considered static. Many of the parameters were assigned somewhat arbitrarily and need to be validated. Other changes of a more fundamental nature may be required due to the findings of future validation studies, and the model should evolve to reflect emerging knowledge of effects such as operator performance following overload (East, 1993; Farrell and Hendy, 1993).

To facilitate changes to the parameters of this implementation, all default values and constants associated with the loading and allocation of attention modules should be set in a single location in the simulation software. It will be possible to revert to the original defaults as required, although a facility should be provided for making a permanent (non-recoverable) change to these values. These data include:

- the interference coefficients of Tables 1 to 5;
- the critical value of the interference coefficient that forces serial processing;
- the percentage complete after which a task can not be interrupted;
- the number of allowable attempts for rescheduling a sheddable task;
- the maximum allowable length of the short term *memory* queue;
- all default values for the task description fields;
- the length of the window for the moving average (or the multiplier for the longest mean time) and the time resolution for plotting the short term memory queue length;
- the default values for continuous tasks (means, coeff. of variation, distribution type max, min etc.); and
- the value of time pressure which defines a point of excessive load.

CONCLUSIONS

This report takes the theoretical framework of a time-based information processing model for the human operator, and derives various relationships and rules necessary for implementing these ideas in a task network simulation environment. This implementation covers both an allocation of attention module for scheduling tasks, and methods for predicting operator load and performance from the resulting task demand. The model that provides the framework for this report has been partially validated; however, within the context of this implementation further validation is necessary. Many of the parameters required by various elements of the implementation have been assigned arbitrarily and need to be verified. The platform for this validation will be the Systems Operator Loading Evaluation (SOLE) software, a collaborative development between the Defence and Civil Institute of Environmental Medicine (DCIEM) and the Canadian Marconi Company.

The approach advocated breaks from recent trends for workload prediction which have been dominated by resource theory models, to focus once again on the time domain, specifically time pressure, as the prime driver of operator processing load and performance. However, in returning to methods which appear at first glance to be similar to traditional *time occupied* models, the pitfalls inherent in these established procedures have been avoided. The framework for the implementation described in this report incorporates aspects of both serial and parallel processing, acknowledges task interference in multi-task situations, and handles both continuous and discrete tasks. Further, by returning to the time domain, model predictions are more readily testable.

The implementation of the IP Model, described in this report, balances the more traditional focus on workload assessment with an emphasis on operator performance and error. This is achieved by tracking the tasks serviced by the allocation of attention module and logging and categorizing the tasks shed, interrupted or postponed. This change of focus seems reasonable for competitive military systems, where the reduction of operator workload is probably not an achievable goal. For such systems, the aim of the designer should be to maximize overall system performance for a given level of expended operator information processing capacity.

ACKNOWLEDGMENTS

I would like to acknowledge the information on CREWCUT, provided by John Lockett of the US Army Research Laboratory, Human Research and Engineering Directorate, Aberdeen Proving Ground, MD, USA, under the auspices of The Technical Cooperation Program (TTCP), Subgroup U Technical Panel 7 (UTP-7). I would also like to thank David Beevis and Ian Mack of DCIEM, for their comments on earlier drafts of this report.

REFERENCES

Aldrich, T. B., Craddock, W., and McCracken, J. H. (1984). A computer analysis to predict crew workload during LHX scout-attack missions: Volume 1 (MDA903-81-C-0504/ASI479-054-I-84(B)). Fort Rucker, AL, USA: United States Army Research Institute Field Unit.

Card, S. K., Moran, T. P., and Newell, A. (Eds.). (1983). *The Psychology of Human-Computer Interaction*. Hillsdale, NJ, USA: Lawrence Erlbaum Associates.

CMC (1992). Aurora Update evaluation: concluding report (DSS Contract No. W8477-9-AC33/01-QF for DND, Ottawa, Canada). Kanata, Ontario, Canada: Canadian Marconi Company.

CMC (1994). System Operator Loading Evaluation: Methodology documentation (SSC Contract W7711-2-7160/01/XSE for DND, Ottawa, Canada). Kanata, Ontario, Canada: Canadian Marconi Company.

Doering, B., and Berheide, W. (1981). A review of simulation languages and their application to manned systems design. In, J. Moraal and K. F. Kraiss (Eds.), *Manned Systems Design: Methods, Equipment, and Applications*. New York, NY, USA: Plenum Press.

East, K. P. (1993). *Operator performance following overload: Is 'Hysteresis' a factor in complex cognitive tasks*. Unpublished Honours Thesis. Waterloo, Ontario: Wilfrid Laurier University.

Farrell, P. S. E., and Hendy, K. C. (1993). Analysis of the hysteresis effect: Short term memory hypothesis. In, *Proceedings of the Human Factors Association of Canada's 26th Annual Conference*. Mississauga, Ontario, Canada: Human Factors Association of Canada. 73-78.

Glenn, F., Cohen, D., Wherry Jr., R., and Carmody, M. (1994). Development and validation of a workload assessment technique for cockpit function allocation. In, K. C. Hendy (Ed.), *Proceedings of the 'Workshop on Task Network Simulation for Human-Machine System Design.'* Washington, DC, USA: The Technical Cooperation Program, Subgroup U, Technical Panel 7. 59-88.

Hendy, K. C. (1994a). An information processing model for workload and performance prediction. In, K. C. Hendy (Ed.), *Proceedings of the 'Workshop on Task Network Simulation for Human-Machine System* Design.' Washington, DC, USA: The Technical Cooperation Program, Subgroup U, Technical Panel 7. 25-45.

Hendy, K. C. (1994b). Survey of national practices in task network simulation for human machine systems. Washington, DC, USA: The Technical Cooperation Program, Subgroup U, Technical Panel 7.

Hendy, K. C., Kobierski, R. D., and Youngson, G. A. (1992). The integration of human factors engineering analyses with manpower and personnel studies. In,

Proceedings of the Workshop on Liveware Integration Needs. Neuilly-sur-Seine, France: Advisory Group for Aerospace Research and Development, Research Study Group 21, Panel 8. 6-1 to 6-11.

Hendy, K. C., Liao, J., and Milgram, P. (1994). Combining time and intensity effects in assessing operator information processing load. *(in preparation).*

Hillier, F. S., and Lieberman, G. J. (1974). *Operations Research (Second Edition).* San Francisco, CA, USA: Holden-Day, Inc.

Jensen, S. (1994). Scenario profiling — using task network simulations of task analysis and workload as the backbone for integrated human factors studies. In, K. C. Hendy (Ed.), *Proceedings of the 'Workshop on Task Network Simulation for Human-Machine System Design.'* Washington, DC, USA: The Technical Cooperation Program, Subgroup U, Technical Panel 7. 89-94.

Kinchla, R. A. (1980). The measurement of attention. In, R. S. Nickerson (Ed.), *Attention and Performance VIII.* Hillsdale, NJ, USA: Lawrence Erlbaum Associates, Publishers. 213-238.

Linton, P. M., Jahns, D. W., and Chatelier, P. R. (1977). Operator workload assessment model: an evaluation of a VF/VA-V/STOL system. In, *Proceedings of the Aerospace Medical Panel Specialist Meeting (AGARD-CPP-216).* Neuilly-sur-Seine, France: Advisory Group for Aerospace Research and Development. A12-1 to A12-11.

Linton, P. M., Plamondon, B. D., Dick, A. O., Bittner Jr., A. C., and Christ, R. E. (1989). Operator workload for military system acquisition. In, G. R. McMillan, D. Beevis, E. Salas, M. H. Strub, R. Sutton and L. van Breda (Eds.), *Application of Human performance Models to System Design.* New York, NY, USA: Plenum Press. 21-45.

Little, R., Dahl, S., Plott, B., Wickens, C., Powers, J., Tillman, B., Davilla, D., and Hutchins, C. (1993). Crew Reduction in Armoured Vehicles Ergonomic Study (CRAVES) (ARL-CR-80). Aberdeen proving Ground, MD, USA: US Army Research Laboratory, Human Research and Engineering Directorate.

Malone, T. B., Kirkpatrick, M., and Kopp, W. H. (1986). Human factors engineering impact of system workload and manning levels. In, *Proceedings of the Human Factors Society's 30th Annual Meeting.* Santa Monica, CA, USA: Human Factors Society. 763-767.

Meister, D. (1985). *Behavioral Analysis and Measurement Methods.* New York, NY, USA: John Wiley and Sons.

MicroAnalysis (1987). *Micro SAINT: user's guide.* Boulder, CO, USA: Micro Analysis and Design, Inc.

Parks, D. L., and Boucek Jr., G. P. (1989). Workload prediction, diagnosis, and continuing challenges. In, G. R. McMillan, D. Beevis, E. Salas, M. H. Strub, R. Sutton and L. van Breda (Eds.), *Applications of Human Performance Models to Systems Design.* New York, NY, USA: Plenum Press.

Rasmussen, J. (1983). Skills, rules and knowledge; signals, signs and symbols and other distinctions in human performance models. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-13(3), 257-266.

Seigel, A. I., and Wolf, J. J. (1969). *Man-machine Simulation Models*. New York, NY, USA: John Wiley and Sons, Inc.

Shaffer, M. T., Hendy, K. C., and White, L. R. (1988). An empirically validated task analysis (EVTA) of low level army helicopter operations. In, *Proceedings of the Human Factors Society's 32nd Annual Meeting*. Santa Monica, CA, USA: Human Factors Society. 178-182.

Wickens, C. D. (1984). *Engineering Psychology and Human Performance*. Columbus, OH, USA: Charles E. Merrill Publishing Company.

Wingert, J. W. (1973). Function interlace modifications to analytic workload prediction. In, K. Cross and J. McGrath (Eds.), *Crewstation Design: An Interagency Conference*. Santa Barbara, CA, USA: Anacapa Sciences. 373-378.

Wortman, D. B., Duket, S. D., Seifert, D. J., Hann, R. L., and Chubb, G. P. (1978). The SAINT user's manual (AMRL-TR-77-62). Wright Patterson Air Force Base, OH, USA: United States Air Force Aerospace Medical Research Laboratories.

APPENDIX 1

DERIVATION OF TASK COMPLETION TIMES UNDER TIME MULTIPLEXING

APPENDIX 1

This Appendix describes the derivation of task completion times for interfering tasks. Assume that multiple, interfering tasks are processed by a combination of time multiplexed processing, together with processing unaffected by interference effects.

Let,

$t_i(\mathrm{s})$ be the start time of task i,

$t_i(\mathrm{e})$ be the ending time of task i, when performed alone,

$t_{ij}(\mathrm{e})$ be the ending time of task i, when performed with task j,

δT be the processing time overlap between the ith and jth task,

$\mathbf{c}_{ij}$ be the proportion of time (δT) that the overlapping tasks share a common processing structure (for simplification, this time is assumed to be evenly distributed throughout the overlap), and

p_i be the probability that in any given time interval, processing resources will be devoted to task i, rather than task j (note that $p_i = (1 - p_j)$).

Suppose,

$$\mathbf{c}_{ij} = \mathbf{c}_{ji},$$
$$T_i = t_i(\mathrm{e}) - t_i(\mathrm{s}),$$
$$T_{ij} = t_{ij}(\mathrm{e}) - t_i(\mathrm{s}), \text{ and}$$
$$t_j(\mathrm{s}) - t_i(\mathrm{s}) \geq 0.$$

Then two cases need to be considered (see Fig. A1):

CASE 1 — in which the processing of one of the tasks (assume it to be task j) is entirely embedded within the processing time of the other (task i); and

CASE 2 — in which the tasks partially overlap (the processing time of task i, remaining after task j starts, is entirely embedded within the processing time of task j).

CASE 1: Tasks Completely Overlap

In any small time interval δt, it is assumed that for $\mathbf{c}_{ij}\delta t$, due to interference effects, time multiplexed processing takes place, while for time $(1 - \mathbf{c}_{ij})\delta t$ processing is unaffected by interference effects. Hence in the interval δt there is

$$(1 - p_i)\mathbf{c}_{ij}\delta t + (1 - \mathbf{c}_{ij})\delta t$$

processed of task j. Therefore, the number of intervals required to fully process task j is

$$\frac{T_j}{(1-p_i)\mathbf{c}_{ij}\delta t + (1-\mathbf{c}_{ij})\delta t},$$

for a total task completion time of

$$\begin{aligned} T_{ji} &= \frac{T_j}{(1-p_i)\mathbf{c}_{ij}\delta t + (1-\mathbf{c}_{ij})\delta t}\delta t \\ &= \frac{T_j}{1-p_i\mathbf{c}_{ij}}. \end{aligned}$$

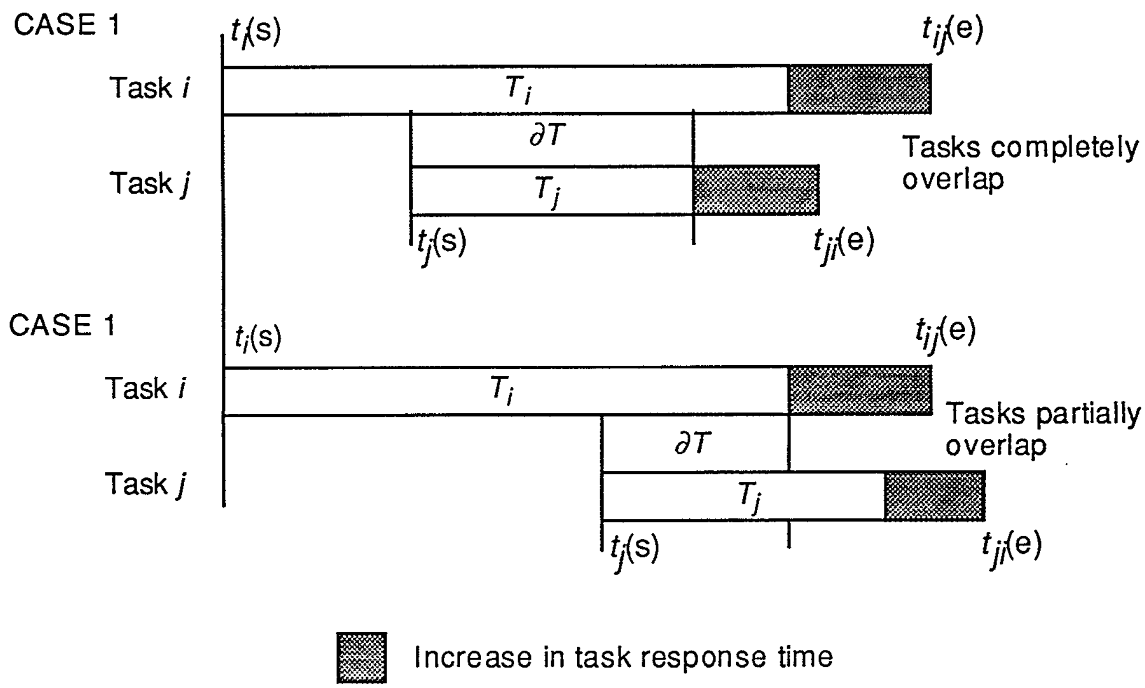


Figure A1. *Two cases of task overlap.*

During the period T_{ji}, $\mathbf{c}_{ij}T_{ji}$ involves time multiplexed processing, while $(1-\mathbf{c}_{ij})T_{ji}$ involves processing unaffected by interference effects. Hence, during the time of overlap between tasks i and j,

$$p_i\mathbf{c}_{ij}T_{ji} + (1-\mathbf{c}_{ij})T_{ji}$$

is processed of task i. Therefore the time to fully process task i is

$$T_{ij} = T_{ji} + \left[T_i - \left\{p_i \mathbf{c}_{ij} T_{ji} + \left(1 - \mathbf{c}_{ij}\right) T_{ji}\right\}\right]$$
$$= T_i + \frac{\mathbf{c}_{ij}\left(1 - p_i\right)}{\left(1 - p_i \mathbf{c}_{ij}\right)} T_j.$$

Sufficient and necessary conditions for CASE 1 are defined by

$$t_j(\mathrm{s}) - t_i(\mathrm{s}) \geq 0, \text{ and}$$

$$t_{ij}(\mathrm{e}) - t_{ji}(\mathrm{e}) \geq 0.$$

Therefore,

$$\left\{t_i(\mathrm{s}) + T_{ij}\right\} - \left\{t_j(\mathrm{s}) + T_{ji}\right\} \geq 0, \text{ or}$$

$$T_i - T_j \left[\frac{1 - \mathbf{c}_{ij}\left(1 - p_i\right)}{1 - p_i \mathbf{c}_{ij}}\right] - \left\{t_j(\mathrm{s}) - t_i(\mathrm{s})\right\} \geq 0.$$

To allow for task interruptions it is necessary to keep a running total of the amount of actual processing time devoted to each task. At any time $t_j(\mathrm{s}) \leq t \leq t_{ji}(\mathrm{e})$, the amount of processing time devoted to task i, since task j commenced, is

$$\Delta T_{ij} = p_i \mathbf{c}_{ij} \left\{t - t_j(\mathrm{s})\right\} + \left(1 - \mathbf{c}_{ij}\right)\left\{t - t_j(\mathrm{s})\right\}$$

and to task j

$$\Delta T_{ji} = \left(1 - p_i\right) \mathbf{c}_{ij} \left\{t - t_j(\mathrm{s})\right\} + \left(1 - \mathbf{c}_{ij}\right)\left\{t - t_j(\mathrm{s})\right\}.$$

Therefore, at time t the amount of processing time remaining on each task is

$$T_i - \sum_j \Delta T_{ij} \text{ for task } i, \text{ and}$$

$$T_j - \sum_i \Delta T_{ji} \text{ for task } j.$$

CASE 2: Tasks Partially Overlap

For CASE 2 (see Fig. A1), not all of task j is processed in concert with task i. However, the processing time of task i $\left(t_i(\mathrm{e}) - t_j(\mathrm{s})\right)$, remaining after the start of task j, is embedded within the processing time of task j. Therefore, CASE 2 is similar to CASE 1, with the partial processing of task i equivalent to task j in CASE 1. Hence the processing time of task i is

$$
\begin{aligned}
T_{ij} &= \left\{t_j(\mathrm{s}) - t_i(\mathrm{s})\right\} + \frac{t_i(\mathrm{e}) - t_j(\mathrm{s})}{1 - p_j \mathbf{c}_{ij}} \\
&= t_j(\mathrm{s}) - t_i(\mathrm{s}) + \frac{t_i(\mathrm{s}) + T_i - t_j(\mathrm{s})}{1 - p_j \mathbf{c}_{ij}} \\
&= \frac{p_j \mathbf{c}_{ij}\left\{t_j(\mathrm{s}) - t_i(\mathrm{s})\right\} + T_i}{1 - p_j \mathbf{c}_{ij}}.
\end{aligned}
$$

Similarly, noting the symmetry with CASE 1, the processing time for task j is

$$
\begin{aligned}
T_{ji} &= \frac{\mathbf{c}_{ij}\left(1 - p_j\right)}{1 - p_j \mathbf{c}_{ij}}\left\{t_i(\mathrm{e}) - t_j(\mathrm{s})\right\} + T_j \\
&= \frac{\left(1 - p_j \mathbf{c}_{ij}\right)T_j + \mathbf{c}_{ij}\left(1 - p_j\right)T_i - \mathbf{c}_{ij}\left(1 - p_j\right)\left\{t_j(\mathrm{s}) - t_i(\mathrm{s})\right\}}{1 - p_j \mathbf{c}_{ij}},
\end{aligned}
$$

noting that $p_j = 1 - p_i$.

For the running total of actual time processed on each task, the equations for CASE 1 apply.

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g., Establishment sponsoring a contractor's report, or tasking agency, are entered in section 12.)
DCIEM, PO Box 2000, North York, Ontario M3M 3B9, CANADA

2. DOCUMENT SECURITY CLASSIFICATION (overall security classification of the document including special warning terms if applicable.)
UNCLASSIFIED

3. DOCUMENT TITLE (the complete document title as indicated on the title page. Its classification should be indicated be the appropriate abbreviation (S,C,R or U) in parentheses after the title.)
IMPLEMENTATION OF AN INFORMATION PROCESSING MODEL FOR TASK NETWORK SIMULATION (U)

4. DESCRIPTIVE NOTES (the category of the document, e.g., technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)
RESEARCH REPORT

5. AUTHOR(S) (Last name, first name, middle initial. If military, show rank, e.g., Burns, Maj. Frank E.)
HENDY, Keith C.

6. DOCUMENT DATE (month and year of publication of document)
MAY 1994

7.a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc.)
46

7.b. NO. OF REFS. (total cited in document)
28

8.a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant)

8.b. CONTRACT NO. (if appropriate, the applicable number under which the document was written)

9.a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by with the document is identified by the originating activity. This number must be unique to this document.)
DCIEM No. 94-40

9.b. OTHER DOCUMENT NO.(S) (any other numbers which may be assigned this document either by the originator or by the sponsor.)

10. DOCUMENT AVAILABILITY (any limitation on further dissemination of the document, other than those imposed by security classification)

X	Unlimited distribution
	Distribution limited to defence departments and defence contractors; further distribution only as approved
	Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved
	Distribution limited to government departments and agencies; further distribution only as approved
	Distribution limited to defence departments; further distribution only as approved
	Other

11. ANNOUNCEMENT AVAILABILITY (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (10.) However, where further distribution (beyond the audience specified in 10) is possible, a wider announcement audience may be selected.)

12. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address.)

UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM
(Highest classification of Title, Abstract, Keywords)

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

Task network simulation is an analytical technique that is widely used to predict operator performance and/or workload during the early stages of systems design. Task network simulation is based on traditional time-line analysis methods, but allows the possibility of non-deterministic task characteristics such as completion times, sequences, outcomes etc. Many simulation environments allow task parameters to vary with various network states, which supports complex logical relationships, and time varying network behaviours.

This report outlines the implementation of a theoretical framework for a new model of the human information processor for use in task network simulation. The development and validation of the Information Processing (IP) Model is described in detail elsewhere. This report deals only with those aspects that are necessary to take the ideas of the IP Model and adapt them for direct application to task network simulation. The material contained in this report provides the bridge between the conceptual descriptions of the IP Model, and the software requirements necessary to put that concept into practice. As part of this process, many parameters are defined and assigned tentative values so that the model can be run within the task network simulation environment.

In devising this implementation, many assumptions were made that are beyond the scope of earlier validation studies. Hence, further validation will be required to verify the model within the context of this implementation. As the specific purpose of the report is to describe the algorithmic and data base requirements for a specific software environment, some of the material is specific to that system. However, much of the material is of a more general nature and could be adapted to other software environments

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible, keywords should be selected from a published thesaurus, e.g. Thesaurus of Engineering and Scientific Terms (TEST) and the thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

HUMAN ENGINEERING
ANALYSIS
TASK NETWORK SIMULATION
SAINT
WORKLOAD
MODELLING
HUMAN INFORMATION PROCESSING
ATTENTION

HFD 07/94

UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM
(Highest classification of Title, Abstract, Keywords)